An Evaluation Resource for Grounding Translation Errors

Anonymous ACL submission

Abstract

The fine-grained manual annotations of translation errors gains more and more attention in machine translation, but these annotations do not ground the errors to the reasons why the annotated text spans are erroneous, resulting in the hardness of evaluating LLMs trustworthiness in the fine-grained error analysis. In this paper, we manually build an evaluation resource for grounding the translation errors through a bi-directional grounding scheme. In the forward direction, we annotate the explanation of reason for each error span. In the backward direction, we annotate the error span given its explanation, in which the error span is masked. If the error spans of both directions are consistent, we deem the explanation is valid. Such grounding process can regulate the explanation so as to avoid the subjective bias. We evaluate LLMs ability in grounding the translation errors on the resource. The results show that LLMs perform significantly worse than human in both directions. Furthermore, we apply the error grounding for filtering false alarmed errors, and achieve significant improvement in translation error detection.

1 Introduction

With the recent development of neural networks and large language models (LLMs), machine translation (MT) systems achieve steady progress in translation quality. Although they perform well in certain circumstances, there still exist various type of errors that need further study. Multidimensional Quality Metrics (MQM) (Lommel et al., 2014a,b) is the fine-grained schema fit for translation error analysis. It contains manual annotation of error spans and has been successfully applied in MT researches on evaluation metrics (Freitag et al., 2021a,b), quality estimation (Zerva et al., 2022), and error correction (Treviso et al., 2024).

Despite its success, MQM annotation only includes information such as error type, location, and severity. There is no manual annotation resource for grounding the translation errors, that is, grounding the errors to the reasons why the annotated text spans are erroneous translations. The scarcity of such resource impedes the interpretability of current researches in error analysis and the building of trustworthy MT models, which should be able to predict the translation errors based on the solid ground of knowing the reason why they are erroneous.

In this paper, we manually build the first resource for grounding the translation errors. Figure 1 illustrates the building process, which adopts a bi-directional grounding scheme (BGS). In the forward grounding, the errors are grounded to their explanations, which state the reason why they are deemed errors. In the backward grounding, the explanations are inversely grounded to the corresponding errors. With error spans masked, the explanations are used for identifying the errors in the translation results. Through BGS, the error and explanation are mutually checked to guarantee their validity, and are adjusted to achieve enhanced consistency.

Based on our manually annotated resource, we establish the evaluation protocol for testing LLMs ability in grounding the translation errors. It shows that LLMs perform significantly worse than human, demonstrating the importance of our annotated resource as the new benchmark for LLMs to enhance the ability in the translation error grounding. Regarding the comparison between the manual explanation and the auto explanation generated by LLMs (Treviso et al., 2024), our manual explanation is more effective for locating the error span than the auto explanation.

Furthermore, we apply translation error grounding for automatically filtering false alarmed errors. Specifically, given automatically detected errors

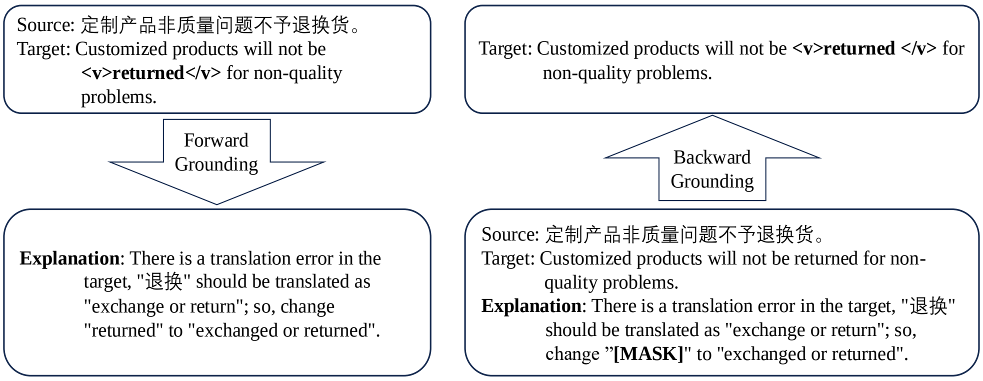


Figure 1: The illustration of BGS for grounding the translation errors. The error spans are annotated between <v> and </v>. In the forward grounding, given the source sentence (source), the translation result (target), and the error span, we annotate the explanation for the error span. In the backward grounding, given the source, the target, and the explanation with the error span masked by '[MASK]', we identify the error span in the target according to the explanation.

(Guerreiro et al., 2024), we filter errors that are not consistent before and after BGS since the false alarmed error may be grounded to a hallucinated explanation, which is in turn grounded to a different text span. Only the errors keeping consistent after BGS are saved as true errors, which have the solid ground of reasons. We found that LLMs with better ability in translation error grounding are more effective for filtering the false alarmed errors. In summary, the contributions of our work are as follows:

- We manually build the first evaluation resource for grounding the translation errors through BGS.
- Different LLMs show different abilities in grounding the translation errors, and they all perform significantly worse than human on the evaluation resource.
- We filter the false alarmed errors by grounding the errors. Through filtering groundless errors, we achieve significant improvement in fine-grained error detection.

2 Related Works

Grounding translation errors is related to the fine-grained error analysis, which is beyond assigning a single sentence-level score for evaluating the translation quality. The fine-grained error analysis focuses on specific error words or phrases, and gradually gains attentions in MT researches. We detail the fine-grained error analysis researches and their relation to the translation error grounding.

2.1 MQM Schema

MQM schema was first introduced in Lommel et al. (2014a,b) as a measurement and analysis framework for the fine-grained MT error analysis. It is adopted in Freitag et al. (2021a,b) for the evaluation metrics task which examines how well an automatic evaluation metric correlates with human judgements. They annotated the fine-grained errors according to the MQM schema, and found that these annotations are more trustworthy for the task. These annotations are subsequently used in the quality estimation task which estimates the quality of MT output without relying on reference translations (Zerva et al., 2022). Due to the success of MQM annotations, they are widely adopted in series of WMT evaluation campaigns, and the annotations are enriched to incorporate more translation results of WMT 2020-2023 submissions[1].

Despite the success of MQM annotations, they do not ground errors to the reasons why they are erroneous, which hampers the building of trustworthy MT models or LLMs. In comparison, we manually create the resource for grounding the translation errors.

2.2 Grounding Translation Errors

Current grounding approaches utilize LLMs to perform fine-grained error analysis, which includes explanations for specific errors. Treviso et al. (2024) use GPT-4 to generate explanations for the errors and use the generated data to fine-tune a multilingual LLM to be able to ground the errors to their explanations. InstructScore is a fine-grained ex-

[1] https://github.com/google/wmt-mqm-human-evaluation

plainable evaluation metric that fine-tunes LLMs to generate quality score accompanied by explanations for the translation errors (Xu et al., 2023; Dandekar et al., 2024). Fine-grained errors and their explanations are also used as prompts for LLMs to refine overall translation results (Treviso et al., 2024; Ki and Carpuat, 2024; Xu et al., 2024; Li et al., 2024).

The explanations in the current grounding approaches are automatically generated by LLMs. There is no manually built resource as a benchmark for evaluating the ability in grounding translation errors. Moreover, the current grounding approaches only use uni-direction grounding, i.e., grounding the errors to the explanation of reasons. In this paper, we establish the benchmark through BGS that bi-directionally checking the errors and the explanations to guarantee their validity.

2.3 Span-level Error Detection

Span-level error detection is crucial for the fine-grained error analysis, which paves the way to the trustworthy MT models. It is achieved by utilizing pre-trained language models or LLMs. AutoMQM uses in-context examples to directly prompt LLMs to identify error spans in the translation results (Fernandes et al., 2023). In contrast to the direct prompts, InstructScore utilizes LLMs to synthesize span-level errors and uses the errors to fine-tune a smaller LLMs to perform the fine-grained error analysis, which includes the span-level error detection (Xu et al., 2023). xCOMET collects available data from translation quality estimation task and metrics task to fine-tune a large encoder model through a multi-task training objective, which includes the span-level error detection (Guerreiro et al., 2024).

Current span-level error detection does not depend on grounding errors to the reasons, which makes the error detection less explainable and groundless. In the mean time, current detectors tend to over-predict errors (Treviso et al., 2024). In comparison, we use BGS to check the authenticity of the errors to filter false alarmed errors.

3 Building The Evaluation Resource for Grounding Translation Errors

We build the resource by manually grounding the MQM annotated errors (Freitag et al., 2021a). Specifically, we select MQM manual annotations on translation results submitted in WMT2022 Chinese-to-English (ZH-EN) and English-to-German (EN-DE) general translation task to ground the errors. This selection contains results of 7 participated teams in Chinese-to-English task and 15 participated teams in English-to-German task. We uniformly select equal number of sentences for each participating team to annotate, and each team do not overlap in the source side. In the end, we have around 2.0K manual grounding instances for each translation direction. Detailed statistics are listed in the appendix A.1.

3.1 Bi-directional Grounding Scheme (BGS)

The resource is built through BGS, which contains three steps:

1. Forward grounding: Given the MQM errors, we annotate the explanations for them explaining why they are erroneous.

2. Backward grounding: Given the explanations with errors masked, we annotate the error spans in the translation result.

3. Calibration: We calibrate the annotations if their forward grounding is not consistent with the backward grounding.

We set different annotators for the different steps to ensure there is no knowledge leakage of grounding answers. Through the mutual checking in BGS, the errors and explanations are regulated to enhanced quality and be consistent with each other to avoid the subjective bias (Treviso et al., 2024).

Forward Grounding. Explanation for each translation error can vary dramatically among different annotators. So we control the explanation annotation through two standards: basic elements and type-specific templates.

The basic elements are basic text spans in the explanation that adequately explain the reason of the errors. For example, in Figure 1, the explanation for the mistranslation contains the informations of the source span ("退换") that is aligned to the error and the correction ("exchanged or returned") of the error. We deem these informations as basic elements in the explanation and categorize them into five categories: source span, target span, error span, correction span, and insertion position. Examples of the five categories are listed in Table 1.

We ask annotators to annotate the basic elements in each explanation unless specific elements are not fit. Source span, target span, error span, and

Source Target Category Severity	定制产品非质量问题不予退换货。 Customized products will not be <v>returned</v> for non-quality problems. Accuracy/Mistranslation Major
Explanation	There is a translation error in the target, "<s>退换</s>" should be translated as "<t>exchange or return</t>"; so, change "<e>returned</e>" to "<a>exchanged or returned".
Source Target Category Severity	多吃，能使您<v>心神</v>安康！ Eat more to make you feel healthy! Accuracy/Omission Major
Explanation	There is no translation for "<e>心神</e>" in the target; so, it should be translated as "<a>both physically and mentally" and added between "healthy" and "!" .

Table 1: Examples of the basic elements in the explanation. Source span is tagged between <s> and </s>, target span is tagged between <t> and </t>, error span is tagged between <e> and </e>, correction span is tagged between <a> and , and insertion position is tagged between <p> and </p>.

Source Target Reference Category Severity	我不等了，取消订单 I won't <v>**wait**</v>. Cancel the order I am done waiting, and I'll cancel the order. Style/Awkward Minor
Explanation	The style of the target does not conform to language conventions, "我不等了" should be translated as "I am done waiting"; so, change "**I won't wait**" to "I am done waiting".

Table 2: Example of the original MQM error span needing the adjustment, which will move '<v>' to the left of 'I' in the target according to the explanation.

	ZH-EN		EN-DE	
	w/o ref.	w/ ref.	w/o ref.	w/ ref.
Perfect Match	87.3	88.5	89.7	90.7
Fuzzy Match	97.1	97.7	97.6	98.0
F1-score	94.4	95.1	94.7	95.2

Table 3: Backward grounding results(%) on identifying the error spans by the annotators.

correction span appear commonly across all error types, while the insertion position only appears in the omission type.

Other than the basic elements, we define the type-specific templates for annotating the explanations. A part of the templates are illustrated in the appendix Table 11. Some error types, such as the mistranslation type, have relatively fixed templates with fixed basic elements, while other types such as grammar errors exhibit rich format due to their flexible reasons.

Backward Grounding. Backward grounding verifies the forward grounding by checking if the error span can be correctly located according to the explanation, which has the error span masked. If the explanation in the forward grounding is valid, the error span will be correctly identified by the annotator. As illustrated in Figure 1, given the source sentence, its translation, and the explanation, the annotators are asked to identify the error span in the translation.

In the backward grounding process, we found some error spans in the original MQM annotations need adjustments on their boundary. For example, the MQM error span tagged between <v> and </v> in Table 2 is "wait", while in the explanation, the error span is extended to "I won't wait" since it should be corrected integrally. For such kind of cases, we adjust the boundaries of the original MQM annotations to consider the correction, and the error spans identified by the annotators in these cases are compared against the adjusted error spans.

Table 3 lists the backward grounding results. 'Perfect Match' denotes the ratio of the error spans identified by the annotators fully matching the MQM annotations (including the adjusted annotations). 'Fuzzy Match' denotes the ratio of the error spans identified by the annotators sharing some parts with the MQM annotations. F1-score (Zerva et al., 2024) evaluates the position match between the error spans identified by the annotators and the MQM annotations. Since the reference translations are not always available, we ask the annotators to identify the error spans without the references at first, then provide the references for the annotators for comparison.

Table 3 shows that providing references moderately enhances the match ratio and F1-score compared to those without the references. This indicates that, based on the explanation, the annotators can identify the error spans for most cases even without the references. It also shows that Fuzzy Match is significantly higher than Perfect Match, indicating that most of the error locations can be identified by the annotators according to the explanation, only the boundaries of the error spans are not correct.

Figure 2 shows the detailed Fuzzy Match results grouped by the different sharing part proportion (spp): spp = (# of sharing characters) / (# of characters of the identified error spans). When spp > 0 is grouped, it is the most loosely fuzzy match that one sharing character is enough for the successfully fuzzy match. When spp > higher threshold is grouped, it becomes more rigorous about the fuzzy match, resulting in lower fuzzy match rate, but the rate is still above 90% in the most rigorous case.

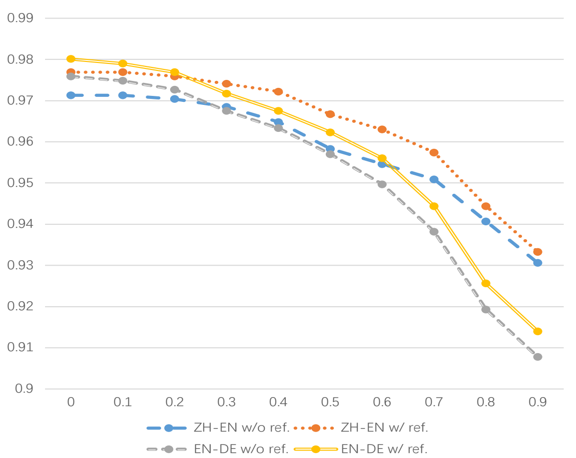


Figure 2: Detailed fuzzy match results. X axis is the spp threshold, Y axis is the fuzzy match rate.

Calibration. After the backward grounding, around ten percent of error spans are not perfectly matched as shown in Table 3. We ask the annotator to re-annotate them, and find that a portion of them (73% in ZH-EN and 25% in EN-DE) can be corrected to be perfectly matched after careful thinking, and the other portion of them (27% in ZH-EN and 75% in EN-DE) can not be corrected due to the invalid explanations. So we refine these invalid explanations until their backward grounding can identify the perfectly matched error spans. After the calibration process, the invalid explanations can be refined, resulting in the overall enhancement of the explanation quality.

4 LLMs Ability in Grounding Translation Errors

Based on our evaluation resource, we test LLMs ability in grounding translation errors. We use the open source Llama3.1-8B-Instruct(Llama3.1 for short) and a proprietary LLM GPT-4 for the testing.

4.1 Evaluation on The Forward Grounding

Given the error, we prompt LLMs to generate the explanation. Besides verifying the explanation through the backward grounding, we evaluate the explanation by checking the basic elements introduced in section 3.1. We regulate the explanation by specific prompts listed in the appendix Table 12.

The generated explanation should contain the basic elements to well explain the error. The accuracy of the basic elements is: acc. = (# of matched basic elements) /(# of total basic elements). In case LLMs generating overlong explanations, we add a brevity penalty: BP = exp(1 - $\frac{l_s}{l_c}$) if ($l_s \geq l_c$), where l_s is the length of the generated explanation, l_c is the length of the human annotated explanation. if ($l_s < l_c$), we set BP = 1. The final evaluation score of the generated explanation is: BP$\times$acc. Because in some circumstances, the references or the error types are not always available, we include the final evaluation score under these conditions in Table 4. It shows that GPT-4 is more accurate than Llama3.1 in both language pairs and all conditions. When references are not available, the performance decreases by a large margin. In comparison, the performance decrease is not so significant when error types are not available.

Figure 3 reports element-wise acc. and type-wise final evaluation score of the generated explanations under the condition of 'ref.+error type'. In type-wise score, we report the top-five frequent error types's score, and the other error types are grouped into one score. In these detailed comparison, GPT-4 exhibits significant advantage over Llama3.1. In short, different LLMs perform differently in the forward grounding, but the performance is not satisfied with the final evaluation score often below 60%.

4.2 Evaluation on The Backward Grounding

Given the manually annotated explanations with error spans masked, we prompt GPT-4[2] to locate the

[2]Llama3.1 does not always maintain the original translation when locating the error spans, while GPT-4 can keep the original translation intact. So we only report GPT-4 performance in locating the error spans.

	ZH-EN		EN-DE	
	Llama3.1	GPT-4	Llama3.1	GPT-4
ref. + error type	0.43	**0.58**	0.53	**0.63**
w/o error type	0.42	**0.54**	0.49	**0.62**
w/o ref.	0.26	**0.44**	0.23	**0.38**
w/o ref. and error type	0.22	**0.36**	0.20	**0.35**

Table 4: The final evaluation score of the basic elements in the forward grounding by LLMs.

	ZH-EN		EN-DE	
	xTower	Manual	xTower	Manual
w/o ref.				
PerfectMatch	15.83	**57.41**	38.47	**57.65**
FuzzyMatch	66.85	**90.93**	78.83	**89.73**
F1-score	40.19	**76.96**	60.67	**76.75**
w/ ref.				
PerfectMatch	17.04	**58.80**	35.85	**52.73**
FuzzyMatch	66.57	**90.83**	77.99	**88.26**
F1-score	40.89	**78.20**	58.65	**73.71**

Table 5: Backward grounding results(%) on identifying the error spans by GPT-4.

	ZH-EN			EN-DE		
All Error Types	PerfectMatch	FuzzyMatch	F1-score	PerfectMatch	FuzzyMatch	F1-score
All Elements	57.41	90.93	76.96	57.65	89.73	76.75
-Correction Span	43.33	86.57	67.13	53.25	89.20	73.81
-Source Span	48.24	89.54	71.57	55.77	89.83	75.39
-Target Span	49.91	90.83	73.36	56.92	90.67	77.90
Omission Error	PerfectMatch	FuzzyMatch	F1-score	PerfectMatch	FuzzyMatch	F1-score
All Elements	44.74	81.58	71.29	47.37	94.74	74.27
-Insertion Position	42.86	81.43	69.88	42.11	89.47	67.69

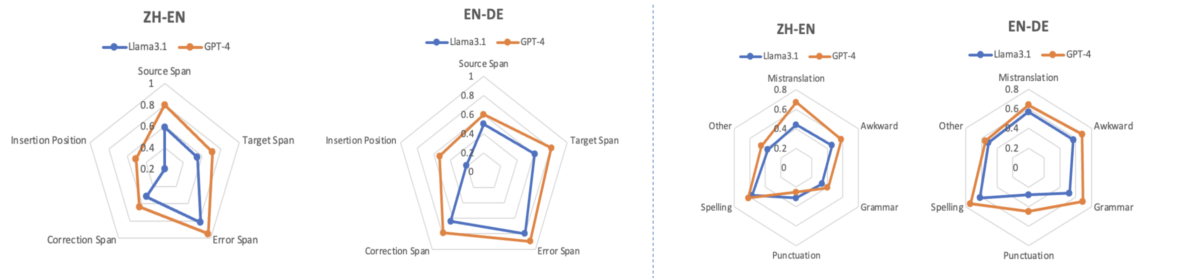
Table 6: The ablation study on the basic elements in the backward grounding.

error spans in the translations with prompts listed in the appendix Table 13. In the mean time, we also include auto explanations generated by LLMs to compare with our manual explanations for the backward grounding. xTower is an LLM fine-tuned on a dataset that includes GPT-4 generated explanations (Treviso et al., 2024). It is used to generate explanations for each error. Table 5 presents the comparison results.

Manual explanation is better than auto explanation. The manual explanation leads a wide margin over the auto explanation generated by xTower. Since the manual explanation is succinct and adequate, while xTower explanation is in free style that scatters attention to the exact error, It is easier for GPT-4 to attend over the manual explanation than over xTower explanation for locating the error spans. Reference effect is marginal or negative in the backward grounding. It is probably because reference contains many information irrelevant to the error, thus distracting GPT-4's attention on locating the error.

LLMs performs worse than human in the backward grounding. When compare Table 5 with Table 3, based on the same manual explanation, GPT-4 locates the errors with perfect match rate below 60%, while human performs with perfect match rate around 90%. This significant difference raises the demand of improving LLMs ability in grounding the errors.

In addition, to test the effectiveness of the basic elements in the explanation, we carry out the ablation study by masking the corresponding basic elements in the explanation (the error span is always masked). The ablation results are presented in Table 6. It shows that the correction span contributes more to the overall performance than the other basic elements. It contains the most helpful information about the error, guiding GPT-4 to easily locate the error span in the translation. Since the basic element of the insertion position only exists in the error type of omission, and the omission error does not happen frequently, we present the performance of the omission error alone at the bottom of Table 6. It shows that the insertion position



(a) (b)

Figure 3: Detailed evaluation on the generated explanations in the forward grounding. (a) is the element-wise acc., and (b) is the type-wise final evaluation score.

is key in the explanation for the omission error. The performance decreases significantly if it is masked.

4.3 Evaluation on The Bi-directional Grounding

In the bi-directional grounding, we let GPT-4 generate explanation in the forward grounding, then reversely identify the error span according to the explanation in the backward grounding. This process is fully automatic, and the error span in the generated explanation is automatically masked by pattern matching in the backward direction. Reference is not used in the bi-directional grounding.

	ZH-EN	EN-DE
PerfectMatch	45.37	47.80
FuzzyMatch	88.80	88.89
F1-score	73.07	71.74

Table 7: The bi-directional grounding results(%) by GPT-4.

The explanation generated by GPT-4 in this process follows the format of the manual explanation. Table 7 lists the error span accuracy after the bi-directional grounding. Compared to Table 5, the bi-directional grounding performs worse than the backward grounding, indicating that GPT-4 explanation is not as effective as the manual explanation for automatically locating the errors. It also shows that this formatted GPT-4 explanation performs better than the free-style xTower explanation. The advantage is more significant in ZH-EN than in EN-DE.

5 Filtering False Alarmed Errors by Error Grounding

BGS is an ecosystem that explains the error in the forward direction, then verifies the explanation in the backward direction. Through such explanation and verification process, true errors will be solidly grounded, while false alarmed errors will hardly find their grounds since they may be grounded to hallucinated explanations in the forward direction, which in turn result in different errors in the backward direction. So, we filter the false alarmed errors by checking whether the error spans remain consistent after BGS, which is executed by LLMs.

5.1 Iterative BGS

We build the error pool by using xCOMET to automatically detect error spans in the translation without using reference (Guerreiro et al., 2024). Since the errors are over-predicted (Treviso et al., 2024), the error pool contains many false alarmed errors needing to be filtered. Considering that a false alarmed error may drift away to a different error span by one iteration of BGS, we propose iterative BGS that iteratively locates the error span until the error span becomes stable, i.e., the error span of the current iteration is the same to that of the last iteration. If an xCOMET error is not consistent with its final error span detected by the iterative BGS, we deem this xCOMET error the false alarmed one and filter it.

The process is presented in algorithm 1, where n is the number of iterations. In each iteration, BGS takes current error as input, and outputs the newly identified error. The iteration ends when the current and new errors are the same or it reaches the maximum number of iterations. Then we compare the final error e'' with the original xCOMET error e through a function named checkConsistency. The function computes the overlap rate, that is, (# of positions shared between e'' and e) / (length of e''), and return true if the rate is above a threshold, meaning that e'' and e are consistent. If this rate is lower than the threshold, then e'' and e have small sharing parts, indicating that e is a false alarmed

Algorithm 1 Iterative BGS

```
for each xCOMET error e do
    e' = e;
    for i = 1 to n do
        e'' =BGS(e')
        if e'' == e' then
            break;
        end if
        e' = e'';
    end for
    if !checkConsistency(e, e'') then
        Filter e;
    end if
end for
```

	ZH-EN	EN-DE
Full Set		
xCOMET	38.4	35.0
IterativeBGS$_{\text{1st}}$	39.1	35.2
IterativeBGS$_{\text{final}}$	**39.5**	**36.1**
Partial Set		
xCOMET	54.5	49.1
IterativeBGS$_{\text{Llama3.1}}$	51.2	44.4
IterativeBGS$_{\text{GPT4}}$	**55.6**	**49.6**

Table 8: F1-score(%) of xCOMET errors and the filtered errors by the iterative BGS, which uses either GPT-4 or Llama3.1 for auto grounding translation errors.

error that causes the error drift, and should be filtered. We set $n = 5$, and the threshold as 0.5 in our experiments.

5.2 Result

We conduct the experiments on our translation error grounding resource. F1-score (Zerva et al., 2024) is used to evaluate the performance by checking position match between human annotated errors and auto detected errors. Table 8 reports the performances. Since GPT-4 and Llama3.1 behave differently in the iterative BGS, that is, Llama3.1 fails in maintaining all original translations when locating the error span (failing rate is 0.42 for ZH-EN and 0.52 for EN-DE), while GPT-4 always keep all original translations unchanged, we divide Table 8 into two parts: One is the performances on the full set, the other is the performances on the partial set that Llama3.1 can maintain the original translations.

In Table 8, IterativeBGS$_{\text{1st}}$ denotes only using the first iteration of the iterative BGS for filtering xCOMET errors, and IterativeBGS$_{\text{final}}$ denotes using the last iteration of the iterative BGS for the filtration. On the full set, both approaches only use GPT-4. It shows that the error grounding by the iterative BGS can effectively filter the false alarmed errors predicted by xCOMET, resulting in significant F1-score improvement. The first iteration leads to better filtration performance, and the iterative process keeps improving the performance over the one pass BGS. The average number of iterations in the iterative BGS is 1.8 for ZH-EN and 1.9 for EN-DE, demonstrating the effectiveness of the iterative algorithm.

On the partial set, Llama3.1 performs significantly worse than xCOMET, while GPT-4 achieves the best performance. It indicates that the ability in grounding translation errors is vital in filtering false alarmed errors. As section 4 evaluates based on our manually built benchmark, GPT-4 has much better performance on grounding translation errors than Llama3.1, both in the forward and the backward directions, leading to the reliable filtration that improves the overall performance. This demonstrates the urgency of improving the LLMs ability in grounding the translation errors. Our evaluation resource can be set as the benchmark for this target.

6 Conclusion

Current manual annotation of the fine-grained translation errors does not explain the reason why they are erroneous, resulting in the hardness of checking whether LLMs trustworthily know the reason when they conduct the fine-grained error analysis. In this paper, we manually build the resource for evaluating LLMs trustworthiness in grounding the translation errors. The bi-directional grounding scheme is proposed for the building. In the forward direction, the errors are manually grounded to their explanations. In the backward direction, the explanations are verified by checking whether the errors can be manually detected according to the explanations, which have the error spans masked. LLMs are evaluated on this resource through such explanation and verification process. Results show that LLMs performs significantly worse than human in both directions. There is large room for LLMs to improve their grounding ability. Furthermore, we apply the error grounding for filtering false alarmed errors, and achieve significant accuracy improvement in the error detection.

Limitations

In our evaluation of LLMs ability in grounding the translation errors, we acknowledge certain limitations in the covered scope. Firstly, our study only evaluates GPT-4 and Llama3.1, not encompassing wide variety of LLMs. This omission represents an area for potential future exploration to provide a more comprehensive understanding of the abilities of various LLMs in the error grounding. Secondly, manual error identification is only conducted for the manual explanations in the backward grounding. LLMs are used instead of the manual method for the error identification when verifying the large volume of the explanations generated by LLMs (LLMs are also used for verifying the manual explanations for fair comparison).

Ethics Statement

We honor the Code of Ethics. We do not use any private data or non-public information in this work. Regarding the manual grounding annotation, we recruit our annotators from the linguistics departments of local universities through public advertisement with a specified pay rate. All annotators are graduate students who took the annotation as a part-time job with salaries above the local basic standard. The annotation does not involve any personally sensitive information.

References

Chinmay Dandekar, Wenda Xu, Xi Xu, Siqi Ouyang, and Lei Li. 2024. Translation canvas: An explainable interface to pinpoint and analyze translation systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 344–350, Miami, Florida, USA. Association for Computational Linguistics.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André Martins, Graham Neubig, Ankush Garg, Jonathan Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1066–1083, Singapore. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Dayeon Ki and Marine Carpuat. 2024. Guiding large language models to post-edit machine translation with error annotations. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4253–4273, Mexico City, Mexico. Association for Computational Linguistics.

Jiahuan Li, Shanbo Cheng, Shujian Huang, and Jiajun Chen. 2024. MT-PATCHER: Selective and extendable knowledge distillation from large language models for machine translation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6445–6459, Mexico City, Mexico. Association for Computational Linguistics.

Arle Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2014a. Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Tradumàtica: tecnologies de la traducció*, 0:455–463.

Arle Richard Lommel, Aljoscha Burchardt, Maja Popovic, Kim Harris, and Hans Uszkoreit. 2014b. Using a new analytic measure for the annotation and analysis of mt errors on real data. In *Proceedings of the 17th Annual Conference of the European Association for Machine Translation. Annual Conference of the European Association for Machine Translation (EAMT-14)*.

Marcos V Treviso, Nuno M Guerreiro, Sweta Agrawal, Ricardo Rei, José Pombal, Tania Vaz, Helena Wu, Beatriz Silva, Daan Van Stigt, and Andre Martins. 2024. xTower: A multilingual LLM for explaining and correcting translation errors. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15222–15239, Miami, Florida, USA. Association for Computational Linguistics.

Wenda Xu, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, Biao Zhang, Zhongtao Liu, William Yang Wang, Lei Li, and Markus Freitag. 2024. LLMRefine: Pinpointing and refining large language models via fine-grained actionable feedback. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1429–1445, Mexico City, Mexico. Association for Computational Linguistics.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5967–5994, Singapore. Association for Computational Linguistics.

Chrysoula Zerva, Frederic Blain, José G. C. De Souza, Diptesh Kanojia, Sourabh Deoghare, Nuno M. Guerreiro, Giuseppe Attanasio, Ricardo Rei, Constantin Orasan, Matteo Negri, Marco Turchi, Rajen Chatterjee, Pushpak Bhattacharyya, Markus Freitag, and André Martins. 2024. Findings of the quality estimation shared task at WMT 2024: Are LLMs closing the gap in QE? In *Proceedings of the Ninth Conference on Machine Translation*, pages 82–109, Miami, Florida, USA. Association for Computational Linguistics.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the WMT 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 69–99, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

A Appendix

A.1 Statistics of The Evaluation Resource for Grounding The Errors

Table 9 and 10 list the statistics according to the error types and severity classes. In the forward grounding, AvgExpLen denotes the average number of words in the explanations, and ElemNo denotes the number of the basic elements shown in Table 1. In the backward grounding, MaskNo denotes the number of the masks in the explanation to avoid the answer leakage, and ChangedErrNo denotes the number of changed MQM annotations such as the one listed in Table 2.

In the translation errors, mistranslation, awkward, grammar, punctuation, and spelling are five major types of the errors. In ZH-EN, major errors and minor errors are equally distributed, while in EN-DE, minor errors take up the majority. The average length of the explanations are around 20-30 words. The number of the basic elements and masks varies along with different error types. After the forward and backward grounding, the original MQM annotations are modified by the careful explanations and verifications. This modification happens more in ZH-EN than in EN-DE.

A.2 The Overlap Rate in Filtering The False Alarmed Errors

Figure 4 shows the performance variance along with the changing overlap rate threshold. The performance peaks when the threshold is set 0.5-0.6, and tends to decrease when the threshold is big. When the threshold is 0.9, the performance is even below the baseline. The curves indicate that if more than a half of an error span identified by the iterative BGS deviates from the xCOMET error, the error tends to be a false alarmed error and should be filtered.

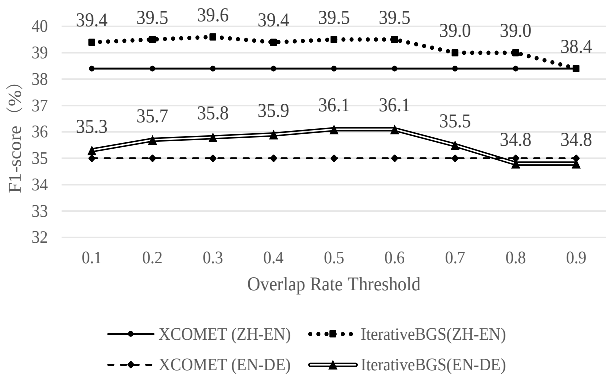


Figure 4: F1-score with different overlap rate threshold in filtering the false alarmed errors.

		ForwardGrounding		BackwardGrounding	
Error Type	No.	AvgExpLen	ElemNo.	MaskNo.	ChangedErrNo.
Mistranslation	532	25.8	2022	532	62
Awkward	145	26.5	464	160	34
Grammar	119	26.1	286	131	14
Punctuation	81	19.2	146	113	1
Spelling	72	31.8	180	72	0
Omission	76	30.5	220	76	0
Addition	17	17.3	17	24	0
Inconsistency	13	28.2	26	13	2
Terminology	9	29.6	36	9	0
Source language fragment	6	16.3	18	6	0
Locale convention	5	22.2	20	5	1
Source error	2	7.0	0	0	2
Non-translation	2	45.0	8	2	1
Register	1	25.0	4	1	0
Severity	No.	AvgExpLen	ElemNo.	MaskNo.	ChangedErrNo.
Major	528	27.0	1954	528	54
Minor	552	25.0	1490	607	63

Table 9: Statistics of ZH-EN resource built by manually grounding the translation errors.

		ForwardGrounding		BackwardGrounding	
Error Type	No.	AvgExpLen	ElemNo.	MaskNo.	ChangedErrNo.
Mistranslation	302	22.4	1178	302	4
Awkward	236	23.2	637	307	1
Grammar	132	26.4	304	158	7
Punctuation	112	19.4	258	123	5
Spelling	43	23.7	143	60	2
Source language fragment	42	14.9	126	42	2
Terminology	23	23.7	92	23	0
Omission	19	26.2	57	19	0
Inconsistency	17	21.7	32	19	0
Register	17	25.6	60	17	0
Addition	5	19.0	10	10	0
Locale convention	3	21.3	12	3	0
Character encoding	2	19.0	4	2	0
Source error	1	7.0	0	0	1
Severity	No.	AvgExpLen	ElemNo.	MaskNo.	ChangedErrNo.
Major	206	21.9	742	206	6
Minor	748	22.9	2169	898	16

Table 10: Statistics of EN-DE resource built by manually grounding the translation errors.

Error Type	Explanation Template
Accuracy/Addition	There is no information about [err] in the source, but it is included in the translation; so, delete [err].
Accuracy/Mistranslation	There is a translation error in the target, [src] should be translated as [tgt]; so, change [err] to [correction].
Accuracy/Omission	There is no translation for [src]; so, it should be translated as [correction] and added [position].
Accuracy/Source language fragment	The translation of [src] in the source is wrong; so, change [err] to [correction].
Fluency/Grammar	There is a grammatical error in the translation ......; so, change [err] to [answer].
Fluency/Inconsistency	There is an inconsistency in the translation, [src] is translated as [err] in the missing context; so, change [err] to [correction].
Fluency/Punctuation	There is a punctuation error in the translation, [src] should be translated as [tgt]; so, change [err] to [correction].
Fluency/Register	There is a fluency issue in the translation that does not fit the context ......; so, change [err] to [correction].
Fluency/Spelling	There is a spelling error in the translation, [err] should be spelled as [tgt]; so, change [err] to [correction].
Fluency/Character encoding	There is a garbled character in the translation; so, change [err] to [correction].
Locale convention	There is a format error in the translation, [src] should be translated as [tgt]; so, change [err] to [correction].
Style/Awkward	The style of the translation does not conform to language conventions ......; so, change [err] to [correction].
Terminology	There is a terminology in the translation that is inappropriate for context, [src] should be translated as [tgt]; so, change [err] to [correction].
Non-translation	It is impossible to reliably characterize distinct errors in the target, [src] should be translated as [tgt]; so, change [err] to [correction].

Table 11: Type-specific templates for the explanations. Slots specified in [ ] should be filled in with the basic elements.

ZH-EN
Please explain why the text labeled between <v> and </v> is a translation error. The format should be consistent with the examples below. The response should be started by "Explanation: ". Do not include any additional analysis or explanations after the correction. Chinese: 圣街通向哪儿? English translation:Where does <v>St. Street</v> lead? Explanation:There is a translation error in the target, "圣街" should be translated as "Sheng Street"; so, change "St. Street" to "Sheng Street". Chinese:当大多数人都以为，英特尔此番举动将对台积电造成一定冲击，并很有可能抢走台积电的“饭碗”。 English translation:When most people thought that Intel’s move would have a certain impact on TSMC, it was very likely to take away TSMC’s “<v>rice bowl</v>”. Explanation:There is a translation error in the target, "饭碗" should be translated as "job" in the context; so, change "rice bowl" to "job". Chinese:实用商务英语口语情景100+ 商务英语口语大百科（附赠多重口语学习赠品） English translation:Practical Business English Speaking Scenarios 100+ Encyclopedia of Business English Speaking (with multiple <v>oral learning gifts</v>) Explanation:The style of the target does not conform to language conventions, "口语学习赠品" should be translated as "gifts for practicing oral English"; so, change "oral learning gifts" to "gifts for practicing oral English". Chinese: {src} English translation: {tgt} Explanation:
EN-DE
Please explain why the text span labeled between <v> and </v> is a translation error. The format should be consistent with the examples below.The response should be started by "Explanation: ". Do not include any additional analysis or explanations after the correction. English:If we did, we’d see these mass gun shootings go down. German translation:Wenn wir das täten, würden wir solche <v>massenhaften Schießereien</v> erleben. Explanation:There is a translation error in the target, "mass gun shootings" should be translated as "viele Amokläufe"; so, change "massenhaften Schießereien" to "viele Amokläufe". English: Also all orders placed on the weekends will be dispatched within the next working days. German translation: Auch alle Bestellungen, die an den Wochenenden <v>platziert</v> werden, werden innerhalb der nächsten Werktage versandt. Explanation: There is a misnomer in the target, "platziert" means putting, and "aufgegeben" means dispatching; so, change "platziert" to "aufgegebenen". English:{src} German translation:{tgt} Explanation:

Table 12: The prompt for the forward grounding using GPT-4.

ZH-EN
Please locate the translation error span in the translation according to the explanation of the error, and do not correct the original translation. The response should be started by "Error Tagging:", and the error span location should be tagged between <v> and </v>. Chinese: 圣街通向哪儿? English translation: Where does St. Street lead? Explanation: There is a translation error in the target, "圣街" should be translated as "Sheng Street"; so, change "[MASK]" to "Sheng Street". Error Tagging:Where does <v>St. Street</v> lead? Chinese: 实用商务英语口语情景100+ 商务英语口语大百科（附赠多重口语学习赠品） English translation: Practical Business English Speaking Scenarios 100+ Encyclopedia of Business English Speaking (with multiple oral learning gifts) Explanation: The style of the target does not conform to language conventions, "口语学习赠品" should be translated as "gifts for practicing oral English"; so, change "[MASK]" to "gifts for practicing oral English". Error Tagging:Practical Business English Speaking Scenarios 100+ Encyclopedia of Business English Speaking (with multiple <v>oral learning gifts</v>) Chinese:{src} English translation:{tgt} Explanation:{exp} Error Tagging:
EN-DE
Please locate the translation error span in the translation according to the explanation of the error, and do not correct the original translation. Note that the error in most cases is masked by "[MASK]" in the explanation. Your task is to recover the error. The response should be started by "Error Tagging:", and the error span location should be tagged between <v> and </v>. English:If we did, we'd see these mass gun shootings go down. German translation: Wenn wir das täten, würden wir solche massenhaften Schießereien erleben. Explanation:There is a translation error in the target, "mass gun shootings" should be translated as "viele Amokläufe"; so, change "[MASK]" to "viele Amokläufe". Error Tagging:Wenn wir das täten, würden wir solche <v>massenhaften Schießereien</v> erleben. English:Also all orders placed on the weekends will be dispatched within the next working days. German translation: Auch alle Bestellungen, die an den Wochenenden <v>platziert</v> werden, werden innerhalb der nächsten Werktage versandt. Explanation:There is a misnomer in the target, "[MASK]" means putting, and "aufgegeben" means dispatching; so, change "[MASK]" to "aufgegebenen". Error Tagging:Auch alle Bestellungen, die an den Wochenenden <v>platziert</v> werden, werden innerhalb der nächsten Werktage versandt. English:{src} German translation:{tgt} Explanation:{exp} Error Tagging:

Table 13: The prompt for the backward grounding using GPT-4.